Heterogeneous Device Discovery Framework for the Smart Homes

Warriach, Ehsan Ullah; Kaldeli, Eirini; Bresser, Jaap; Lazovik, Alexander; Aiello, Marco

Published in:
IEEE GCC Conference Exhibition

Publication date:
2011

HETEROGENEOUS DEVICE DISCOVERY FRAMEWORK FOR THE SMART HOMES

Ehsan Ullah Warriach, Eirini Kaldeli, Jaap Bresser, Alexander Lazovik, Marco Aiello

Distributed Systems Group, Johann Bernoulli Institute
University of Groningen, Nijenborgh 9, 9747 AG, The Netherlands
{e.u.warriach,e.kaldeli,j.bresser,a.lazovik,m.aiello}@rug.nl

ABSTRACT

The future home will be pervaded by mobile and stationary devices which need to dynamically connect and coordinate in order to seamlessly help people in accomplishing their tasks. However, for this vision to become a reality, it is important for researcher and developer to build applications that constantly adapt and integrate heterogeneous devices supported by different communication technologies. Bluetooth efficiently fits in the home environment: the range of communication it provides sufficient for Smart Homes applications consisting of a large number of resource-constraint devices. In this paper, we present a Pervasive Discovery Framework for Smart Homes, and an implementation currently supporting the Bluetooth and UPnP device discovery. The framework is based on robust technologies such as the UPnP protocol and the Java based OSGi bundles.

Index Terms— Bluetooth discovery, Pervsive Discovery Framework, Smart Homes, UPnP, OSGi

1. INTRODUCTION

Pervasive computing in various application domains typically focused on basic system integration such as interconnecting sensors, actuators, computers and other devices in the environment. In the context of smart homes, pervasive computing calls for the deployment of a wide variety of intelligent devices in working and living environments. Devices react to their environments and coordinate with each other and with networked services. In order to guarantee safety and comfort for the inhabitants, the devices should be able to react to contextual changes and coordinate with each other and with networked services. Furthermore, many devices are expected to be mobile, and dynamically discover other devices at a given location and continue to function even if they are disconnected. The overall goal is to provide users with information, assist them with performing complex tasks, and appropriately react to situations that need to be dealt with. For example, consider a scenario where a disabled user on a wheel-chair is preparing food in the kitchen, when suddenly the smoke detector identifies a potentially dangerous smoke. As a result, a predefined goal for taking care of this situation is automatically triggered: the user is safely moved out of the kitchen, the kitchen door is closed to isolate the smoke in the kitchen, the ventilator is turned on, and a mobile alarm curried by the user's nurse is turned on to notify him about the smoke.

A pervasive computing environment can thus be envisioned as a combination of mobile and stationary devices that draw on powerful services embedded in the network to achieve users tasks. The result is a system, with tens of thousands of people, devices, and services coming and going. The key challenge for developers is to build applications that adapt to such a highly dynamic environment and continue to function even if people and devices are roaming across the Smart Home and if the network provides only limited services.

Smart Homes are dynamic and technologically heterogeneous. Several layers of operation governed established standards, home networks and middleware equipped with different communication protocols are put together to enable automation and efficient management of the Smart home. Unfortunately, many pervasive computing systems lack the ability to evolve as new technologies emerge. Integrating numerous heterogeneous elements is mostly a manual, ad-hoc process. Inserting a new element requires investigating its characteristics and operations, determining how to configure and integrate it, while repeated testing is necessary to avoid conflicts in the overall system. To facilitate this tedious process, standardized protocols and technologies are needed, as well as an automatic approach to discovering heterogeneous devices and services. Examples of such protocols include Universal Plug and Play (UPnP) [4], the Open Services Gateway initiative (OSGi) framework [3], Jini [2] and Bluetooth Service Discovery Protocol (SDP) [1].

Bluetooth is a commonly used technology with low-power consumption, cheap hardware interfaces and short range. Bluetooth is used in a wide variety of customer-focused devices, most of which are capable of running Java. Java provides the APIs for the Bluetooth Wireless technology, in order to facilitate development of Bluetooth-

The research is supported by the EU project Smart Homes for All (http://www.sm4all-project.eu), contract FP7-224332.

enabled applications on a Java platform. Bluetooth can be used as a low level communication technology, which in our case connects a large number of resource-constrained devices to a central Home Gateway.

As far as home networking architecture is concerned, UPnP has been used for over a decade [10] with the aim of allowing devices to connect easily to each other, and simplifying the implementation of home networks. In order to meet the requirement for platform and device independence, we are using the OSGi framework [7]. Additional network stacks, such as Jini [2], ZigBee [5] and Konnex [8], can also be supported. Long Ngo [12] describes the challenges that currently exist in interconnecting home devices and recognizes that OSGi can be useful for developing smart homes. In the followings, we describe an architecture to solve the discovery and interoperation challenges in dynamic and open environments by using standardized technologies (OSGi, UPnP).

This paper is structured as follows. Section 2 provides an overview of the proposed architecture, while Section 3 explains the Pervasive Discovery framework, the Bluetooth discovery and the UPnP discovery frameworks. Section 4 provides details of the implemented RuG-ViSi visualization tool. A conclusion is given and future research is specified in section 5.

2. THE SMART HOMES FOR ALL PROJECT

In the context of the EU project Smart Homes for ALL (SM4ALL) [13] an architecture for the smart home is proposed whose main components are illustrated in Figure 1. It consists of three layers. On top is the user layer which provides the interfaces to the home such as touch screen and a Brain-Computer Interface for disabled people [6]. The composition layer contains the logic to manage and control the devices, satisfy users desired goals and support eventing. The composition layer details can be found in [9]. At the bottom is the pervasive layer, where the heterogeneous actuators, sensors and mobile devices of the house live. Existing communication between software and hardware components at pervasive layer is illustrated in Figure 1. A Device can be anything from a Bluetooth phone or UPnP media center to motorized blinds. The Home Gateway is the one where we install all the pervasive layer software. It can be a personal computer (PC) with several interfaces and communication capabilities, where we can install and deploy our software.

The pervasive layer is a service-oriented event-driven software layer. Device and service discovery as well as interoperability are important requirements for pervasive home networking, and as described in [7], these can be efficiently met by using a centralized gateway. The OSGi [3] framework is a platform and device independent framework that can be used to develop service gateways. Therefore we choose to use OSGi as a fundamental component of our pervasive layer. Figure 2 illustrates the current implementation of the pervasive framework.

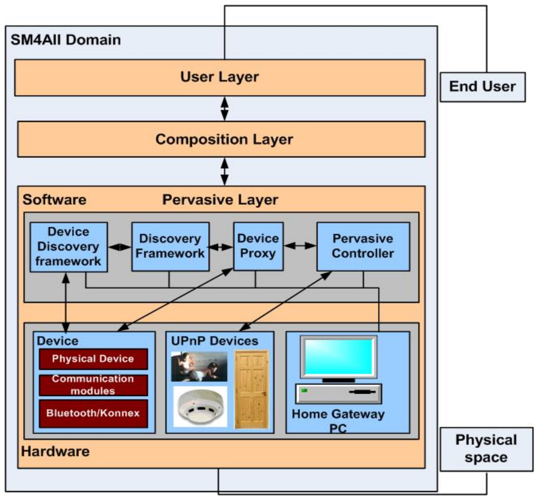

Figure 1. Architecture Overview of SM4ALL Domain

3. PERVASIVE DISCOVERY FRAMEWORK

Heterogeneous devices can connect and disconnect dynamically with respect to the SM4ALL middleware, and services of connected devices are offered to the rest of the system. In order to be able to integrate heterogeneous devices into the SM4ALL middleware, the pervasive framework provides an extensible Pervasive Discovery Framework (PDF). This discovery framework transforms the devices proprietary representation into a common representation (UPnP), that can be understood by the rest of the middleware layers. The development of an extensible pervasive discovery framework of heterogeneous devices poses already many challenges, including scalability, security and dynamicity. The major requirements that the PDF has to fulfill include device discovery, device-services discovery, service control, eventing, platform independence, extensibility and good performance.

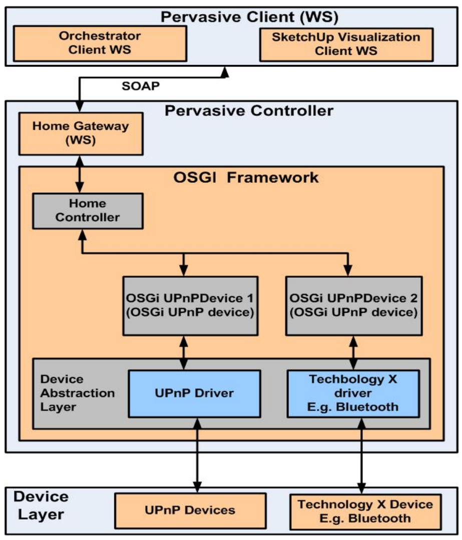

Figure 2. Architecture of Pervasive Framework

To satisfy these PDF requirements, we use two standardized technologies, namely, UPnP and OSGi. The UPnP standard defines a set of networking protocols based on established standards such as TCP/IP and UDP, and offers support for discovery, description, control, event notification and presentation [4]. OSGi is the wrapping layer that bridges UPnP and Web Service level service invocations. UPnP devices can join or leave, as well as non-UPnP devices such as Bluetooth, through the deployment of appropriate drivers. It is possible to support other technologies by writing additional drivers. Figure 2 provides an overview of the internal architecture of the pervasive layer.

The pervasive controller at the pervasive layer is a so-called OSGi bundle that is responsible for handling eventing and control of the UPnP services available in the OSGi framework. It functions as a bridge between the OSGi layer and the Web services layer. The OSGi framework provides a Java interface to define and interact with UPnP devices, and additional features such as life cycle management for services and dependency management.

The device abstraction layer in Figure 2, abstracts away the underlying device technology layer above it. For each communication technology (Bluetooth, UPnP, ZigBee) that the system supports, a driver must be available. The responsibility of a driver is to wrap devices as instances of the UPnPDevice[1] interface and register them as OSGi services. This allows the home controller to communicate transparently with heterogeneous devices.

3.1. Bluetooth Discovery Framework

To be able to plug in a new Bluetooth-enabled device into the SM4ALL network, we need to implement a general spontaneous networking mechanism. There needs to be at least a single device (Home Gateway) in the Smart Home system as shown in Figure 1. The Home Gateway is loaded with a Bluetooth Adapter (BT discovery). This device is called base node in the context of Bluetooth discovery, and the Home Gateway will always have a Bluetooth server listener for incoming connections within the home network.

Bluetooth discovery is a part of the PDF. It is always active and scanning for devices, and establishes communication with the middleware layer. When a new device appears in the SM4ALL network, it extracts all the information the device is capable of providing, such as the name, type and the unique Bluetooth 48-bit address of the device. The information gathered during the Bluetooth physical discovery process, is used for reasoning in the device ontology to infer the category of the device and the services it provides. The result of the query is a link to a proxy (OSGi bundle) that handles the communication with the device and offers a UPnP interface. The proxy is installed and initiated in the proxies container component (OSGi Framework) as shown in Figure 2. The proxy is paired with the physical discovered Bluetooth device. The device proxy is discovered by the pervasive controller (Figure 2), enabling the device to interoperate in the framework of the SM4ALL middleware. Exceptions are raised in cases where devices cannot be discovered, e.g., when the device type is unknown, or the device cannot be resolved with the information provided.

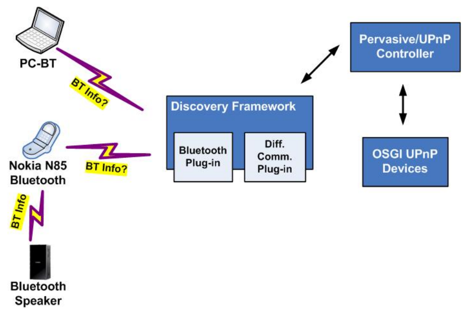

Figure 3. Bluetooth Discovery Framework

For two kinds of devices the above architecture is implemented: a desktop system with Java Standard Edition and a Java Mobile Edition (J2ME) running on a mobile phone (Nokia N85). We have split the SM4ALL Home Gateway software part into a Bluetooth generic interface library and a Bluetooth device discovery specific part (BluetoothDiscovery). A simulated UPnP Lamp Device is registered to the UPnP network as OSGi bundles, to test the Bluetooth discovery module as illustrated in Figure 4. A Bluetooth device that has been discovered is registered in the SM4ALL middleware. When the device is properly configured, it is installed at the OSGi framework. Thus, a Lamp UPnP representation can control the discovered physical Bluetooth device.

3.2. UPnP Discovery

The pervasive controller is an extended UPnP control point for UPnP devices as shown in the Figure 2 and provides a common access for the upper middleware layer. UPnP provides support for automatic discovery and communication with device-services. The UPnP protocol is based on discovery, device description, control and event notification.

4. SIMULATION AND VISUALIZATION

Testing and verifying the behaviour of Pervasive Discovery Framework in large pervasive systems is costly, setup time in a real home is high, and tuning the physical devices consumes a great deal of time. Therefore, an environment that mimics as closely as possible the real setting and is able to simulate a number of interactions and behaviours can greatly help the development and testing of the proposed Pervasive Discovery Framework with different communication technologies. It also serves the purpose of acceptability testing with potential users of the system.

The goal is to create a simulation of a home in which home applications are integrated. The aim is to show that

[1] org.osgi.service.upnp.UPnPDevice (see `http://www.osgi.org/javadoc/r4v42/org/osgi/service/upnp/UPnPDevice.html`)

all heterogeneous networks and devices (sensors and actuators) with different services and communication technologies (e.g., Bluetooth, KNX, UPnP, ZigBee) seamlessly integrate into the middleware, provided that the supported services use a common and standard abstraction interface. For instance Figure 4 depicts a home consisting of a living room and kitchen.

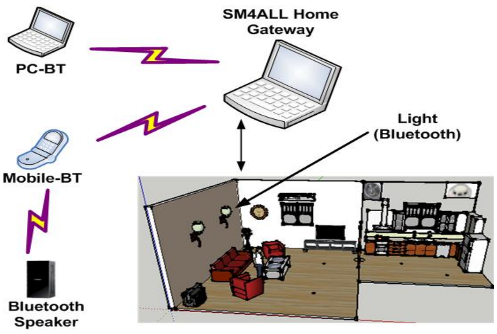

Figure 4. Simulation Environment

To test and verify the behaviour of Bluetooth and UPnP discovery modules, we have improved our initial implementation of a visualization tool for home environments [11] —the RuG ViSi tool—, based on Google SketchUp. We have upgraded it to a Bluetooth Discovery module compliant with the SM4ALL architecture, which is capable of full bidirectional interactions with UPnP services or devices deployed with a UPnP proxy. The visualization module is registered as a client of the server on top of the OSGi framework. In this way, one can control the devices at the pervasive layer through their virtual equivalents, while the invocations of the actual devices are in turn reflected at the visualization layer, by changing its state. The RuG ViSi tool can be naturally coupled with real physical devices. For instance, we have coupled a light service with a Bluetooth enabled mobile and desktop PC. The hardware is plugged in the OSGi layer and turns on and off a specific light in the virtual home as shown in Figure 4.

After the bootstrap of the middleware, the Bluetooth's discovery framework turns on and continuously searches for devices near SM4ALL Home Gateway. The PC can control the Lamp in the RuG ViSi tool by sending the on and off command through SM4ALL Home gateway, after successfully establishing a Bluetooth connection with the Home Gateway.

The architecture of the RuG ViSi tool is an instance of the general one proposed in [9] as shown in the Figure 2, it is mainly covering the pervasive layer and orchestrator. The orchestrator module, we use in the current demo to derive the simulation and visualization is registered as a client, and is responsible for registering the state of the home, as well as for coordinating the available services.

5. CONCLUDING REMARKS

The proposed pervasive framework is highly dynamic, extendable and open, in which devices join and leave while adopting different communication technologies. The current implementation supports UPnP and Bluetooth communications. As a next step, we plan to extend it to support other communication technologies such as ZigBee, Konnex and BACNet. The main difference with existing efforts it that our proposal is to integrate these technologies into a simple and comprehensive framework targeted at the pervasive computing applications.

We also plan to evaluate the system with potential home users who are clients of a Dutch home care organization (Thuiszorg Het Friese Land - THFL). Clients with motoric disabilities will use the Brain-Computer Interface (BCI), while elder users will use a touch screen. Example of scenarios will include the satisfaction of goals such as a request to watch TV. The results of these test will be used to evaluate the performance and to make recommendations to improve the SM4All system. A video illustrating the SM4All RuG ViSi tool video is available at `http://www.youtube.com/watch?v=2w_UIwRqtBY`.

6. REFERENCES

[1] Blueooth service discovery p., `www.bluetooth.org`

[2] Jini, `www.jini.org`

[3] OSGi service platform core spec. v.4, `www.osgi.org`

[4] Upnp™ device architecture version 1.1, `www.upnp.org`

[5] ZigBee, `www.zigbee.org`

[6] Babiloni, F., Cincotti, F., Marciani, M., Astolfi, L., Aloise, F., Fallani, F., Mattia, D.: On the use of bci outside scientific laboratories: Toward an app. in domotic environments. In: Int. Review of Neurobiology (2009)

[7] Dobrev, P., Famolari, D., Kurzke, C., Miller, B.A.: Device and service discovery in home networks with osgi. Communications Magazine, IEEE 40(8), 86–92 (2002)

[8] Jammes, F., Mensch, A., Smit, H.: Service-oriented device communications using the devices profile for web services. In: Int. Ws. on Middleware for pervasive and ad-hoc computing (MPAC '05). pp. 1–8. ACM (2005)

[9] Kaldeli, E., Warriach, E.U., Bresser, J., Lazovik, A., Aiello, M.: Interoperation, composition and simulation of services at home. In: ICSOC (2010). Springer: To appear

[10] Kim, D.S., Lee, J.M., Kwon, W.H., Yuh, I.K.: Design and implementation of home network systems using upnp middleware for networked appliances. IEEE Transactions on Consumer Electronics pp. 963–972 (2002)

[11] Lazovik, E., den Dulk, P., de Groote, M., Lazovik, A., Aiello, M.: Services inside the smart home: A simulation and visualization tool. In: ICSOC. pp. 651–652 (2009)

[12] Ngo, L.: Service-oriented architecture for home networks. In: Seminar on Internetworking. pp. 1–6 (2007)

[13] SM4All: Smart hoMes for All. http://www.sm4all-project.eu (2008-2011)